The Metamath Proof Language

Norman Megill
May 9, 2014

Metamath development server
Overview of Metamath

- Very simple language: substitution is the only basic rule

- Very small verifier ($\approx$300 lines code)

- Fast proof verification (6 sec for $\approx$18000 proofs)

- All axioms (including logic) are specified by user

- Formal proofs are complete and transparent, with no hidden implicit steps
Goals

Simplest possible framework that can express and verify (essentially) all of mathematics with absolute rigor

Permanent archive of hand-crafted formal proofs

Elimination of uncertainty of proof correctness

Exposure of missing steps in informal proofs to any level of detail desired

Non-goals (at this time)

Automated theorem proving

Practical proof-finding assistant for working mathematicians
<table>
<thead>
<tr>
<th>Contributors</th>
</tr>
</thead>
<tbody>
<tr>
<td>David Abernethy</td>
</tr>
<tr>
<td>Stefan Allan</td>
</tr>
<tr>
<td>Juha Arpiainen</td>
</tr>
<tr>
<td>Jonathan Ben-Naim</td>
</tr>
<tr>
<td>Gregory Bush</td>
</tr>
<tr>
<td>Mario Carneiro</td>
</tr>
<tr>
<td>Paul Chapman</td>
</tr>
<tr>
<td>Scott Fenton</td>
</tr>
<tr>
<td>Jeffrey Hankins</td>
</tr>
<tr>
<td>Anthony Hart</td>
</tr>
</tbody>
</table>
Examples of axiom systems expressible with Metamath
(Blue means used by the set.mm database)

- Intuitionistic, **classical**, paraconsistent, relevance, quantum propositional logics

- Free or **standard** first-order logic with equality; modal and provability logics

- NBG, **ZF**, NF set theory, with **AC**, GCH, **inaccessible** and other large cardinal axioms

Axiom schemes are **exact** logical equivalents to textbook counterparts. All theorems can be instantly traced back to what axioms they use.
What has been accomplished? (1 of 2)

24 of Freek Wiedijk’s “Formalizing 100 Theorems” (from ZFC)

<table>
<thead>
<tr>
<th>√2 irrationality</th>
<th>Cantor’s theorem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Denumerability of rationals</td>
<td>Sum of a geometric series</td>
</tr>
<tr>
<td>Pythagorean theorem</td>
<td>Sum of an arithmetic series</td>
</tr>
<tr>
<td>Euler’s gen of Fermat’s Little Thm</td>
<td>GCD algorithm</td>
</tr>
<tr>
<td>Infinitude of primes</td>
<td>Mathematical induction</td>
</tr>
<tr>
<td>De Moivre’s theorem</td>
<td>Cauchy-Schwarz inequality</td>
</tr>
<tr>
<td>Uncountability of reals</td>
<td>Intermediate value theorem</td>
</tr>
<tr>
<td>Schroeder-Bernstein thm</td>
<td>Fundamental thm of arithmetic</td>
</tr>
<tr>
<td>Binomial theorem</td>
<td>Desargues’s theorem</td>
</tr>
<tr>
<td>Number of subsets of a set</td>
<td>Triangle inequality</td>
</tr>
<tr>
<td>Bezout’s theorem</td>
<td>Bertrand’s Postulate</td>
</tr>
<tr>
<td>Sum of recipr. of triang. numbers</td>
<td>Formula for Pythagorean triples</td>
</tr>
</tbody>
</table>
What has been accomplished? (2 of 2)

Other examples (all proved directly from ZFC axioms)

Hartogs’ theorem (without using Axiom of Choice)
Konig’s theorem (set theory)
Dedekind-cut construction of reals
Pocklington’s theorem (primality test)
Euler’s identity $e^{i\pi} = -1$ (and other complex trig and logs)
Cayley’s theorem
Bolzano-Weierstrass theorem
Heine-Borel theorem
Banach fixed point theorem
Baire’s category theorem
Uniform boundedness principle (Banach-Steinhaus theorem)
Riesz representation theorem
**Theorem bpos**

**Description:** Bertrand's postulate: there is a prime between $N$ and $2N$ for every positive integer $N$. This proof follows Erdős's method, for the most part, but with some refinements due to Shigenori Tochiri to save us some calculations of large primes. See [http://en.wikipedia.org/wiki/Proof_of_Bertrand%27s_postulate](http://en.wikipedia.org/wiki/Proof_of_Bertrand%27s_postulate) for an overview of the proof strategy. (Contributed by Mario Carneiro, 14-Mar-2014.)

**Assertion**

<table>
<thead>
<tr>
<th>Ref</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>bpos</td>
<td>$\vdash (N \in \mathbb{N} \rightarrow \exists p \in \mathbb{P} \ (N &lt; p \land p \leq (2 \cdot N)))$</td>
</tr>
</tbody>
</table>

Distinct variable group: $N, p$

**Proof of Theorem bpos**

<table>
<thead>
<tr>
<th>Step</th>
<th>Hyp</th>
<th>Ref</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>nre 7758</td>
<td>$3 \vdash (N \in \mathbb{N} \rightarrow N \in \mathbb{R})$</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>2re 7809</td>
<td>$4 \vdash 2 \in \mathbb{R}$</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>6nn 7861</td>
<td>$5 \vdash 6 \in \mathbb{N}$</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>nnnn0i 7796</td>
<td>$4 \vdash 6 \in \mathbb{N}_0$</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>reexpcl 6716</td>
<td>$4 \vdash ((2 \in \mathbb{R} \land 6 \in \mathbb{N}_0) \rightarrow (2 \mid 6) \in \mathbb{R})$</td>
</tr>
<tr>
<td>6</td>
<td>2, 4, 5</td>
<td>mp2</td>
<td>Closure of exponentiation of reals.</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>leltic 7278</td>
<td>$3 \vdash ((N \in \mathbb{R} \land (2 \mid 6) \in \mathbb{R}) \rightarrow (N \leq (2 \mid 6) \lor (2 \mid 6) &lt; N))$</td>
</tr>
<tr>
<td>8</td>
<td>1, 6, 7</td>
<td>sylancl 720</td>
<td>$2 \vdash (N \in \mathbb{N} \rightarrow (N \leq (2 \mid 6) \lor (2 \mid 6) &lt; N))$</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>bpos1 13819</td>
<td>$3 \vdash ((N \in \mathbb{N} \land N \leq (2 \mid 6)) \rightarrow \exists p \in \mathbb{P} \ (N &lt; p \land p \leq (2 \cdot N)))$</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>eqid 2075</td>
<td>$\vdash (n \in \mathbb{N} \Rightarrow (((2)^n / 3) \cdot ((x \in \mathbb{R}^+ \Rightarrow ((\log'x) / x))'(\sqrt{n}))) + ((9 / 4) \cdot ((x \in \mathbb{R}^+ \Rightarrow ((\log'x) / x))''(\sqrt{n}))) + ((9 / 4) \cdot ((x \in \mathbb{R}^+ \Rightarrow ((\log'x) / x))'(\sqrt{n}))) + ((9 / 4) \cdot ((x \in \mathbb{R}^+ \Rightarrow ((\log'x) / x))''(\sqrt{n}))) + ((9 / 4) \cdot ((x \in \mathbb{R}^+ \Rightarrow ((\log'x) / x))'(\sqrt{n}))) + ((9 / 4) \cdot ((x \in \mathbb{R}^+ \Rightarrow ((\log'x) / x))''(\sqrt{n}))))$</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>eqid 2075</td>
<td>$\vdash (x \in \mathbb{R}^+ \Rightarrow ((\log'x) / x)) = (x \in \mathbb{R}^+ \Rightarrow ((\log'x) / x))$</td>
</tr>
</tbody>
</table>
Ghilbert

- Ghilbert and Metamath are sister languages. It’s easy to convert between them.
- Modularization: Proofs are organized into files which are imported and exported into other files.
- Online Editor. Proofs can be edited online and have LaTeX typesetting.

Go to: ghilbert-app.appspot.com/wiki/tutorial/overview

Sum of an Arithmetic Series

Proof

\[ \sum_{x=0}^{y} x = \frac{y(y+1)}{2} \rightarrow \sum_{x=0}^{y+1} x = \frac{y(y+1)}{2} + (y+1) \]

\[ \rightarrow \sum_{x=0}^{y+1} x = \frac{(y+2)(y+1)}{2} \]

Detach the last number in a sum: y + 1

Distributive Property

Induction

\[ \sum_{x=0}^{A} x = \frac{A(A+1)}{2} \]
The Quadratic Equation

The quadratic equation gives two possible solutions to a second-order polynomial equation. This proof begins with the assumption that solutions to the equation exists and that the constant $a$ is not 0. If the value of $a$ were 0, the equation would be linear not quadratic.

**Proof**

\[ ax^2 + bx + c = 0 \]
\[ x^2 + \frac{bx}{a} = -\frac{c}{a} \]

Starting Hypothesis

\[ x^2 + \frac{bx}{a} + \left( \frac{b}{2a} \right)^2 = \left( \frac{b}{2a} \right)^2 - \frac{c}{a} \]

Subtract C, Divide A

\[ \left( x + \frac{b}{2a} \right)^2 = \frac{b^2 - 4ac}{(2a)^2} \]

Complete the Square

\[ x + \frac{b}{2a} = \frac{\sqrt{b^2 - 4ac}}{2a} \quad \text{or} \quad x + \frac{b}{2a} = -\frac{\sqrt{b^2 - 4ac}}{2a} \]

Factor the polynomial

Two solutions when inverting a square

\[ x = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad \text{or} \quad x = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \]

Subtract B/2A from both sides

This and other interesting proofs are available at: ghilbert-app.appspot.com/wiki/tutorial/sampler
The Metamath language
Metamath language syntax

Syntax elements: *symbols* (math symbols), *labels* (statement identifiers), and 11 language keywords: $c$ $v$ $f$ $e$ $d$ $a$ $p$ $=$ $\{ \}$

- Constant declaration: $c$ symbols $.$
- Variable declaration: $v$ symbols $.$
- Variable-type assignment: label $f$ symbols $.$
- Logical hypothesis: label $e$ symbols $.$
- Distinct variable proviso: $d$ symbols $.$
- Axiom scheme: label $a$ symbols $.$
- Theorem scheme and its proof: label $p$ symbols $=$ labels $.$
- Delimit scope of $f$, $d$, $e$: $\{ \ldots \}$

Complete specification is in *Metamath* book, pp. 92–95

Proof of Theorem id1

<table>
<thead>
<tr>
<th>Step</th>
<th>Hyp</th>
<th>Ref</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ax-2</td>
<td>5</td>
<td>( \vdash ((\varphi \rightarrow ((\varphi \rightarrow \varphi) \rightarrow \varphi)) \rightarrow ((\varphi \rightarrow (\varphi \rightarrow \varphi)) \rightarrow (\varphi \rightarrow \varphi)) )</td>
</tr>
<tr>
<td>2</td>
<td>ax-1</td>
<td>4</td>
<td>( \vdash (\varphi \rightarrow ((\varphi \rightarrow \varphi) \rightarrow \varphi)) )</td>
</tr>
<tr>
<td>3</td>
<td>1, 2</td>
<td>ax-mp 7</td>
<td>( \vdash ((\varphi \rightarrow (\varphi \rightarrow \varphi)) \rightarrow (\varphi \rightarrow \varphi)) )</td>
</tr>
<tr>
<td>4</td>
<td>ax-1</td>
<td>4</td>
<td>( \vdash (\varphi \rightarrow (\varphi \rightarrow \varphi)) )</td>
</tr>
<tr>
<td>5</td>
<td>3, 4</td>
<td>ax-mp 7</td>
<td>( \vdash (\varphi \rightarrow \varphi) )</td>
</tr>
</tbody>
</table>

Metamath's web page display of id1 proof
Example - Intuitionistic implicational calculus (1 of 2)

Declare 5 constants
$c \vdash \text{wff ( ) -> .}$

Declare 3 variables ($\varphi, \psi, \chi$)
$v \text{ ph ps ch .}$

Establish variable type for $\varphi$
$\text{wph } f \text{ wff ph .}$

Establish variable type for $\psi$
$\text{wps } f \text{ wff ps .}$

Establish variable type for $\chi$
$\text{wch } f \text{ wff ch .}$

Syntax builder for implication
$\text{wi } a \text{ wff ( ph -> ps ) .}$

Two axiom schemes and rule of modus ponens:

$\text{ax-1 } a \vdash ( \text{ ph -> ( ps -> ph ) ) .}$
$\text{ax-2 } a \vdash ( ( \text{ ph -> ( ps -> ch ) ) }$

$\text{-> ( ( ph -> ps ) -> ( ph -> ch ) ) ) .}$

${$

$\text{maj } e \vdash ( \text{ ph -> ps ) .}$

$\text{min } e \vdash \text{ ph .}$

$\text{ax-mp } a \vdash \text{ ps .}$

$}$
Example - Intuitionistic implicational calculus (2 of 2)

Theorem scheme: Identity law

\( \text{id1} \ p \vdash (\ p \rightarrow \ p ) = \)

\[ \begin{array}{cccccccccccc}
\text{wph} & \text{wph} & \text{wph} & \text{wi} & \text{wi} & \text{wph} & \text{wph} & \text{wph} & \text{wi} & \text{wi} & \text{wph} \\
\text{wph} & \text{wph} & \text{wph} & \text{wi} & \text{wi} & \text{wph} & \text{wph} & \text{wph} & \text{wi} & \text{wi} & \text{wph} \\
\text{wph} & \text{wph} & \text{wph} & \text{wi} & \text{wi} & \text{wph} & \text{wph} & \text{wph} & \text{wi} & \text{wi} & \text{wph} \\
\text{wph} & \text{wph} & \text{wph} & \text{wi} & \text{wi} & \text{wph} & \text{wph} & \text{wph} & \text{wi} & \text{wi} & \text{wph} \\
\text{ax-2} \\
\text{wph} & \text{wph} & \text{wph} & \text{wi} & \text{ax-1} & \text{ax-mp} & \text{wph} & \text{wph} & \text{ax-1} & \text{ax-mp} \\
\end{array} \]

Logic step actions and resulting proof steps:

Push ax-2  \( \vdash (\ (\ p \rightarrow (\ (\ p \rightarrow \ p ) \rightarrow \ p ) ) \rightarrow (\ (\ p \rightarrow (\ p \rightarrow \ p ) ) \rightarrow (\ p \rightarrow \ p ) ) ) \)

Push ax-1  \( \vdash (\ p \rightarrow (\ (\ p \rightarrow \ p ) \rightarrow \ p ) ) \)

Pop maj, pop min, push ax-mp

\( \vdash (\ (\ p \rightarrow (\ p \rightarrow \ p ) ) \rightarrow (\ p \rightarrow \ p ) ) \)

Push ax-1  \( \vdash (\ p \rightarrow (\ p \rightarrow \ p ) ) \)

Pop maj, pop min, push ax-mp

\( \vdash (\ p \rightarrow \ p ) \)
“Hidden” hypotheses for substitution assignments to variables in $a$ and $p$ statements

ax-1 showing all hypotheses (pops 2 from stack, pushes 1):

wph $f$ wff ph $.$
wps $f$ wff ps $.$
ax-1 $a \vdash ( \text{ph} \to ( \text{ps} \to \text{ph} ))$ $.$

ax-mp showing all hypotheses (pops 4 from stack, pushes 1):

wph $f$ wff ph $.$
wps $f$ wff ps $.$
min $e \vdash \text{ph}$ $.$
maj $e \vdash ( \text{ph} \to \text{ps} )$ $.$
ax-mp $a \vdash \text{ps}$ $.$
Syntax-building steps for substitution assignments

Theorem scheme: Identity law

\[ \text{id1 } \vdash ( \text{ph} \rightarrow \text{ph} ) = \]

\[
\text{wph wph wph wi wi wph wph wph wi wph wph wi wph wph wph wi wi wph wph wph wi wi wph wph wph wi wph wph wph wi wph wph wph wi wph ax-2 wph wph wph wi ax-1 ax-mp wph wph wph ax-1 ax-mp } .
\]

MM> show proof id1 /all /lemmon

... 

31 wph \text{f wff ph} 
32 wph \text{f wff ph} 
33 wph \text{f wff ph} 
34 32,33 wi \text{a wff ( ph -> ph )} 
35 31,34 ax-1 \text{a } \vdash ( \text{ph} \rightarrow ( ( \text{ph} \rightarrow \text{ph} ) \rightarrow \text{ph} ) ) 

...
Why explicit syntax-building steps?

Theorem scheme: Identity law

\[ \text{id1 } p \vdash ( \text{ ph } \rightarrow \text{ ph } ) = \]
\[ \text{ wph wph wph wi wi wph wph wph wph wi wph wi wph wph wi wph wph wi wi wph wph wph wi wi wph wph wph wph wi wph ax-2 wph wph wph wi ax-1 ax-mp wph wph ax-1 ax-mp } . \]

Only the logic steps “ax-2 ax-1 ax-mp ax-1 ax-mp” are needed theoretically (and by some verifiers e.g. Metamath Solitaire)

Advantages of explicit syntax-building steps:

- Faster verification (no unification needed)
- Simpler verifier (no unification algorithm needed)

Disadvantage:

- Verbose proofs
Compressed proofs

Identity law with **compressed proof**

\[ \text{id1 } \vdash ( \text{ph } \rightarrow \text{ph } ) = ( \text{wi } \text{ax-2 ax-1 ax-mp } ) \text{AAABZBZFAFABBGFBAFACAFDEAADE } \].

Specification is in Appendix B of *Metamath* book

**Advantages:**
- 85% proof size reduction on average (7× smaller)
- 6× faster verification (reading compressed format directly)
- set.mm size breakdown: 8.5MB for proofs, 16.3MB total
Predicate calculus with equality
Classical propositional calculus

We will implicitly assume predicate calculus axioms include:

<table>
<thead>
<tr>
<th>Axiom</th>
<th>Scheme</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Axiom Simp</td>
<td>ax-1</td>
<td>( \varphi \rightarrow (\psi \rightarrow \varphi) )</td>
</tr>
<tr>
<td>Axiom Frege</td>
<td>ax-2</td>
<td>( ((\varphi \rightarrow (\psi \rightarrow \chi)) \rightarrow ((\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \chi))) )</td>
</tr>
<tr>
<td>Axiom Transp</td>
<td>ax-3</td>
<td>( ((\neg \varphi \rightarrow \neg \psi) \rightarrow (\psi \rightarrow \varphi)) )</td>
</tr>
<tr>
<td>Rule of Modus Ponens</td>
<td>ax-mp</td>
<td>( \varphi &amp; \ (\varphi \rightarrow \psi) \Rightarrow \psi )</td>
</tr>
</tbody>
</table>

Axiom schemes for classical propositional calculus
(Łukasiewicz’s system, called \( P_2 \) by Church)
Variables vs. metavariables

Elements of **actual** first-order logic (for set theory):
- Fixed set of individual variables: $v_1, v_2, v_3, \ldots$
- Wffs (well-formed formulas) constructed from variables connected by $=$ and $\in$, which are then used to build up larger wffs connected with $\rightarrow$, $\neg$, $\forall$ (e.g. $(v_1 = v_3 \rightarrow \neg\forall v_2 \ v_2 \in v_4)$).
- There are **no** wff variables

Elements of **Metamath** (set.mm database):
- Individual metavariables $x, y, \ldots$ ranging over $v_1, v_2, v_3, \ldots$
- Wff metavariables $\varphi, \psi, \ldots$ ranging over wffs such as $v_2 \in v_4$ and $(v_1 = v_3 \rightarrow \neg\forall v_2 \ v_2 \in v_4)$
- $x = y$, $x \in y$, $\neg \varphi$, $(\varphi \rightarrow \psi)$, and $\forall x \varphi$ are wff schemes
- Actual variables $v_1, v_2, \ldots$ are **never** mentioned explicitly
Simple schemes and simple metalogic

**Simple scheme** - An axiom scheme or theorem scheme containing only:
1. Wff metavariables $\varphi, \psi, \ldots$ with no arguments
2. Individual metavariables $x, y, \ldots$
3. Provisos of the form “where $x$ and $y$ are distinct”
4. Provisos of the form “where $x$ does not occur in $\varphi$”

**Proof using simple metalogic** - A proof in which each step is a simple scheme—either a direct substitution into an axiom scheme (inheriting any provisos) or an inference rule applied to previous steps.
Proofs: logic vs. simple metalogic

In a standard first-order logic proof, each step is a single instance of an axiom scheme (or rule applied to previous steps) using $v_1, v_2, \ldots$. There are no provisos associated with any step (or the final theorem). All variables are “distinct” by definition.

In simple metalogic, each proof step is itself a scheme using $x, y, \ldots$ and $\varphi, \psi, \ldots$ and possible distinct-variable provisos.
Predicate calculus (with equality) in Metamath

The Metamath language (simple schemes) does not have “free variable” and “proper substitution” as built-in primitives. Traditional predicate calculus cannot be represented directly.

Tarski’s system S2 (1965) (with predicates $=$ and $\in$) is equivalent but has only simple schemes for its axioms.

<table>
<thead>
<tr>
<th>Axiom of Quantified Implication</th>
<th>$\vdash (\forall x(\varphi \rightarrow \psi) \rightarrow (\forall x\varphi \rightarrow \forall x\psi))$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule of Generalization</td>
<td>$\varphi \rightarrow \vdash \forall x\varphi$</td>
</tr>
<tr>
<td>Axiom of Equality (1)</td>
<td>$\vdash (x = y \rightarrow (x = z \rightarrow y = z))$</td>
</tr>
<tr>
<td>Axiom of Existence</td>
<td>$\vdash \neg \forall x \neg x = y$, where $x$ is distinct from $y$</td>
</tr>
<tr>
<td>Axiom of Equality (2)</td>
<td>$\vdash (x = y \rightarrow (x \in z \rightarrow y \in z))$</td>
</tr>
<tr>
<td>Axiom of Equality (3)</td>
<td>$\vdash (x = y \rightarrow (z \in x \rightarrow z \in y))$</td>
</tr>
<tr>
<td>Axiom of Quantifier Introduction</td>
<td>$\vdash (\varphi \rightarrow \forall x\varphi)$, where $x$ does not occur in $\varphi$</td>
</tr>
</tbody>
</table>

Tarski’s system S2
Example of proof as intended by Tarski’s system S2:

1. \( \vdash (v_2 \in v_1 \rightarrow (v_2 = v_1 \rightarrow v_2 \in v_1)) \)  
   \hspace{1cm} \text{ax-1}

2. \( \vdash \forall v_1 (v_2 \in v_1 \rightarrow (v_2 = v_1 \rightarrow v_2 \in v_1)) \)  
   \hspace{1cm} 1, \text{ax-gen}

3. \( \vdash (\forall v_1 (v_2 \in v_1 \rightarrow (v_2 = v_1 \rightarrow v_2 \in v_1))) \)  
   \hspace{1cm} \rightarrow (\forall v_1 v_2 \in v_1 \rightarrow \forall v_1 (v_2 = v_1 \rightarrow v_2 \in v_1))) \)  
   \hspace{1cm} \text{ax-5}

4. \( \vdash (\forall v_1 v_2 \in v_1 \rightarrow \forall v_1 (v_2 = v_1 \rightarrow v_2 \in v_1)) \)  
   \hspace{1cm} 2, 3, \text{ax-mp}

Proof using simple metalogic (Metamath):

1. \( \vdash (\varphi \rightarrow (\psi \rightarrow \varphi)) \)  
   \hspace{1cm} \text{ax-1}

2. \( \vdash \forall x (\varphi \rightarrow (\psi \rightarrow \varphi)) \)  
   \hspace{1cm} 1, \text{ax-gen}

3. \( \vdash (\forall x (\varphi \rightarrow (\psi \rightarrow \varphi))) \rightarrow (\forall x \varphi \rightarrow \forall x (\psi \rightarrow \varphi))) \)  
   \hspace{1cm} \text{ax-5}

4. \( \vdash (\forall x \varphi \rightarrow \forall x (\psi \rightarrow \varphi)) \)  
   \hspace{1cm} 2, 3, \text{ax-mp}
Metalogical completeness

A set of axiom schemes is **metalogically complete** when all valid simple schemes are provable with simple metalogic.

**Example:** System $P_2$ of classical propositional calculus is metalogically complete.

**Problem:** Tarski’s system S2, while *logically* complete, is not *metalogically* complete.

**Example:** $\vdash (x = y \rightarrow (\forall y \varphi \rightarrow \forall x (x = y \rightarrow \varphi)))$ (ax-11 in set.mm) can only be proved in S2 by induction on formula length of $\varphi$

**Solution:** Extend Tarski’s S2 with additional (though logically redundant) simple schemes.
<table>
<thead>
<tr>
<th>ax-5</th>
<th>( \vdash (\forall x(\varphi \rightarrow \psi) \rightarrow (\forall x \varphi \rightarrow \forall x \psi)) )</th>
<th>( \vdash (\forall x(\varphi \rightarrow \psi) \rightarrow (\forall x \varphi \rightarrow \forall x \psi)) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>ax-6</td>
<td>( \vdash (\neg \forall x \varphi \rightarrow \forall x \neg \forall x \varphi) )</td>
<td>( \vdash (\forall x \neg \psi \rightarrow \exists y \forall x \varphi) )</td>
</tr>
<tr>
<td>ax-7</td>
<td>( \vdash (\forall x \forall y \varphi \rightarrow \forall y \forall x \varphi) )</td>
<td>( \vdash (\forall x \neg \psi \rightarrow \exists y \forall x \varphi) )</td>
</tr>
<tr>
<td>ax-gen</td>
<td>( \vdash \varphi \vdash \forall x \varphi )</td>
<td>( \vdash \varphi \vdash \exists x \forall x \varphi )</td>
</tr>
<tr>
<td>ax-8</td>
<td>( \vdash (x = y \rightarrow (x = z \rightarrow y = z)) )</td>
<td>( \vdash (x = y \rightarrow (x = z \rightarrow y = z)) )</td>
</tr>
<tr>
<td>ax-9</td>
<td>( \vdash \neg \forall x \neg x = y )</td>
<td>( \vdash \neg \forall x \neg x = y ), where ( x ) is distinct from ( y )</td>
</tr>
<tr>
<td>ax-10</td>
<td>( \vdash (\forall x x = y \rightarrow \forall y y = x) )</td>
<td>( \vdash (\forall x x = y \rightarrow \forall y y = x) )</td>
</tr>
<tr>
<td>ax-11</td>
<td>( \vdash (x = y \rightarrow (\forall y \varphi \rightarrow \forall x (x = y \rightarrow \varphi))) )</td>
<td>( \vdash (x = y \rightarrow (\forall y \varphi \rightarrow \forall x (x = y \rightarrow \varphi))) )</td>
</tr>
<tr>
<td>ax-12</td>
<td>( \vdash (\neg \forall z z = x \rightarrow (\forall z z = y \rightarrow (x = y \rightarrow \forall z x = y))) )</td>
<td>( \vdash (\neg \forall z z = x \rightarrow (\forall z z = y \rightarrow (x = y \rightarrow \forall z x = y))) )</td>
</tr>
<tr>
<td>ax-13</td>
<td>( \vdash (x = y \rightarrow (x \in z \rightarrow y \in z)) )</td>
<td>( \vdash (x = y \rightarrow (x \in z \rightarrow y \in z)) )</td>
</tr>
<tr>
<td>ax-14</td>
<td>( \vdash (x = y \rightarrow (z \in x \rightarrow z \in y)) )</td>
<td>( \vdash (x = y \rightarrow (z \in x \rightarrow z \in y)) )</td>
</tr>
<tr>
<td>ax-17</td>
<td>( \vdash (\varphi \rightarrow \forall x \varphi), \text{ where } x \text{ does not occur in } \varphi )</td>
<td>( \vdash (\varphi \rightarrow \forall x \varphi), \text{ where } x \text{ does not occur in } \varphi )</td>
</tr>
</tbody>
</table>
Metalogical completeness

**Theorem.** The extended set of axiom schemes ax-1 through ax-17 is **metalogically complete** (Theorem 9.7 in Megill 1995).

**Open problem:** The (metalogical) **independence** of these schemes has not been proven, except for ax-9 and ax-11.

- Independence of ax-9 proved by Raph Levien (2005)
Distinct variable provisos

The axiom scheme “$(\varphi \rightarrow \forall x \varphi)$, where $x$ does not occur in $\varphi$” is expressed in the Metamath language as

$\{$

\begin{align*}
&d \ x \ \text{ph} \\
&ax-17 \ a \ |- \ ( \text{ph} \rightarrow A. \ x \ \text{ph})$
\end{align*}

$\} \$

Rule: Substitutions inherit distinct variable provisos.

Example: Substitute $y = z$ for $\varphi$. Then

$$(\varphi \rightarrow \forall x \varphi)$$

becomes

$$(y = z \rightarrow \forall x \ y = z), \text{ where } x \text{ is distinct from } y \text{ and } z.$$

31
Traditional logic notions using Metamath

Traditional logic: “where $x$ is not free in $\varphi$”
Metamath: use logical ($\varepsilon$) hypothesis $\vdash (\varphi \rightarrow \forall x \varphi)$

Traditional logic: “The proper substitution of $y$ for $x$ in $\varphi$”
Metamath: $[y/x] \varphi$, defined $((x = y \rightarrow \neg \varphi) \rightarrow \forall x (x = y \rightarrow \varphi))$)

Traditional logic: “$\varphi(y)$ where $y$ is free for $x$ in $\varphi(x)$”
Metamath: $[y/x] \varphi$
Definitions in Metamath

- Definitions are introduced as axioms ($\alpha$) and are indistinguishable from axioms to the verifier.

- Soundness (eliminability and non-creativity) depends highly on the underlying logic and cannot be automatically checked generally.

- In set.mm we require new definitions to be automatically checkable. **All but 3 definitions in set.mm are automatically verifiable with a simple algorithm.**
Definitions for predicate calculus in \textit{set.mm}

Definitions extend \textit{wff} syntax, and the definiendum (l.h.s.) and definiens (r.h.s.) are connected with the biconditional $\leftrightarrow$.

\textbf{Examples:}

\textbf{df-an} \quad \vdash ((\varphi \land \psi) \leftrightarrow \neg(\varphi \rightarrow \neg \psi))

\textbf{df-ex} \quad \vdash (\exists x \varphi \leftrightarrow \neg \forall x \neg \varphi)

\textbf{df-eu} \quad \vdash (\exists! x \varphi \leftrightarrow \exists y \forall x (\varphi \leftrightarrow x = y))

where $x$ and $y$ are distinct and $y$ does not occur in $\varphi$

Any new variable on r.h.s. must be distinct from all others.
ZFC set theory
Axiom schemes for ZFC set theory in \textit{set.mm}

<table>
<thead>
<tr>
<th>Axiom of Extensionality</th>
<th>ax-ext</th>
<th>$\vdash (\forall z(z \in x \leftrightarrow z \in y) \rightarrow x = y)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Axiom of Replacement</td>
<td>ax-rep</td>
<td>$\vdash (\forall w \exists y \exists z(\forall y \varphi \rightarrow z = y) \rightarrow \exists y \exists z(z \in y \leftrightarrow \exists w(w \in x \land \forall y \varphi)))$</td>
</tr>
<tr>
<td>Axiom of Power Sets</td>
<td>ax-pow</td>
<td>$\vdash \exists y \exists z(\forall w(w \in z \rightarrow w \in x) \rightarrow z \in y)$</td>
</tr>
<tr>
<td>Axiom of Union</td>
<td>ax-un</td>
<td>$\vdash \exists y \exists z(\exists w(z \in w \land w \in x) \rightarrow z \in y)$</td>
</tr>
<tr>
<td>Axiom of Regularity</td>
<td>ax-reg</td>
<td>$\vdash (\exists y \exists y y \in x \rightarrow \exists y(y \in x \land \forall z(z \in y \rightarrow \neg z \in x)))$</td>
</tr>
<tr>
<td>Axiom of Infinity</td>
<td>ax-inf</td>
<td>$\vdash \exists y(x \in y \land \forall z(z \in y \rightarrow \exists w(z \in w \land w \in y)))$</td>
</tr>
<tr>
<td>Axiom of Choice</td>
<td>ax-ac</td>
<td>$\vdash \exists y \forall z \forall w((z \in w \land w \in x) \rightarrow \exists v \forall u(\exists t((u \in w \land w \in t) \land (u \in t \land t \in y)) \leftrightarrow u = v)))$</td>
</tr>
</tbody>
</table>

All individual metavariables $x$, $y$, $z$, ... below are assumed to be mutually distinct.
Axioms vs. axiom schemes again

In Metamath, every axiom, theorem, and proof step is a simple scheme.

In **standard ZFC set theory**, the Axiom of Extensionality is a **specific axiom** in the language of first-order logic:

\[(\forall v_3 (v_3 \in v_1 \leftrightarrow v_3 \in v_2) \rightarrow v_1 = v_2)\]

In **Metamath (set.mm)**, this is stated as an **axiom scheme**:

\[(\forall z (z \in x \leftrightarrow z \in y) \rightarrow x = y), \text{ where } x, y, z \text{ are distinct}\]

Under first-order logic, every instance of this scheme is logically equivalent to the specific axiom.
Axiom Scheme of Replacement

In set.mm, Replacement is \textit{automatically} a scheme:

\[(\forall w \exists y \forall z (\forall y \varphi \rightarrow z = y) \rightarrow \exists y \forall z (z \in y \leftrightarrow \exists w (w \in x \land \forall y \varphi))),\]

where \(x, y, z, w\) are distinct

By using \(\forall y \varphi\) instead of \(\varphi\), we “protect” it against the case where \(\varphi\) might be substituted with an expression containing \(y\).

Alternately, we could use just \(\varphi\) and add the proviso “where \(y\) does not occur in \(\varphi\).” \textbf{A matter of taste.}

We can also eliminate \textbf{all} provisos:

\[\exists x (\exists y \forall z (\varphi \rightarrow z = y) \rightarrow \forall z (\forall y z \in x \leftrightarrow \exists x (\forall z x \in y \land \forall y \varphi)))]}
A **class builder** is an expression of the form \( \{x \mid \varphi \} \). Let \( A, B, \ldots \) be metavariables ranging over class builders. We extend wffs with the following “definitions:”

\[
\begin{align*}
y \in \{x \mid \varphi \} & \iff [y/x] \varphi \\
A = B & \iff \forall x (x \in A \iff x \in B) \\
A \in B & \iff \exists x (x = A \land x \in B)
\end{align*}
\]

where \( x \) does not occur in \( A \) or \( B \). Soundness (eliminability, non-creativity) must be proved outside of Metamath, and Metamath treats them (like all definitions) as **axioms**.

We can prove \( x = \{y \mid y \in x\} \) when \( x \) and \( y \) are distinct, so an individual variable \( x \) is a special case of a class expression.
Defining new classes

In definitions extending class syntax, the definiendum (l.h.s.) and definiens (r.h.s.) are connected with equality $\equiv$.

**Examples:** Universal class, union of a class, maps-to notation

\[
\text{df-v} \quad \vdash V = \{ x \mid x = x \}
\]

\[
\text{df-uni} \quad \vdash \bigcup A = \{ x \mid \exists y (x \in y \land y \in A) \}
\]
where $x$ and $y$ are distinct and do not occur in $A$

\[
\text{df-mpt} \quad \vdash (x \in A \mapsto B) = \{ (x, y) \mid (x \in A \land y = B) \}
\]
where $x$ and $y$ are distinct, and $y$ does not occur in $A$ or $B$
Emulating deductions in a Hilbert-style system (1 of 2)

- Metamath is intended for **Hilbert-style deductive systems** (axiom schemes plus inference rules)

- **Metamath does not have the Deduction Theorem built in** ("Δ ∪ \{P\} ⊢ Q implies Δ ⊢ P → Q").

- Alternative: Natural deduction emulation
Theorem pockthg

Description: The generalized Pocklington's theorem. If $N - 1 = A \cdot B$ where $B < A$, then $N$ is prime if and only if for every prime factor $p$ of $A$, there is an $x$ such that $x \uparrow (N - 1) = 1 \mod N$ and $\gcd (x \uparrow ((N - 1) / p) - 1, N) = 1$. (Contributed by Mario Carneiro, 3-Mar-2014.)

Hypotheses

<table>
<thead>
<tr>
<th>Ref</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>pockthg.1</td>
<td>$\vdash (\varphi \rightarrow A \in \mathbb{N})$</td>
</tr>
<tr>
<td>pockthg.2</td>
<td>$\vdash (\varphi \rightarrow B \in \mathbb{N})$</td>
</tr>
<tr>
<td>pockthg.3</td>
<td>$\vdash (\varphi \rightarrow B &lt; A)$</td>
</tr>
<tr>
<td>pockthg.4</td>
<td>$\vdash (\varphi \rightarrow N = ((A \cdot B) + 1))$</td>
</tr>
</tbody>
</table>
| pockthg.5 | $\vdash (\varphi \rightarrow \forall p \in \mathbb{P} \ (p \parallel A \rightarrow \exists x \in \mathbb{Z} \ ((x \uparrow (N - 1)) \mod N) = 1$
           | $\land (((x \uparrow ((N - 1) / p)) - 1) \gcd N) = 1))$               |

Assertion

<table>
<thead>
<tr>
<th>Ref</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>pockthg</td>
<td>$\vdash (\varphi \rightarrow N \in \mathbb{P})$</td>
</tr>
</tbody>
</table>
The End

Thank you!
Supplementary slides
Recursive definitions (1 of 2)

Recursive definitions are hard to eliminate. Instead, we can define a “recursive definition generator” (df-rdg):

\[ \vdash \text{rec}(F, A) = \bigcup \{ f \mid \exists x \in \text{On} (f \text{ Fn } x \wedge \forall y \in x f \upharpoonright y = (g \mapsto \text{if}(g = \emptyset, A, \text{if} (\text{Lim dom } g, \bigcup \text{ran } g, \text{F}'(g \cup \text{dom } g)))'((f \upharpoonright y))\}, \]

where \( x, y, f, g \) don’t occur in \( F \) or \( A \)

\( F \) is the characteristic function, \( A \) is the initial value, and \( \text{rec}(F, A) \) is a function on the (proper) class of all ordinals.
Recursive definitions (2 of 2)

Ordinal addition is defined with a direct definition (df-oadd):

\[ \vdash +_o = (x \in \text{On}, y \in \text{On} \mapsto (\text{rec}((z \in V \mapsto \text{suc }z), x) \check{\downarrow} y)) \]

where \( x, y, z \) are distinct

Recursive definition emerges as theorems (oa0, oasuc, oalim):

\[ \vdash (A \in \text{On} \rightarrow (A +_o \emptyset) = A) \]
\[ \vdash ((A \in \text{On} \land B \in \text{On}) \rightarrow (A +_o \text{suc } B) = \text{suc } (A +_o B)) \]
\[ \vdash ((A \in \text{On} \land B \in \text{On} \land \text{Lim } B) \rightarrow (A +_o B) = \bigcup x \in B(A +_o x)), \]

where \( x \) doesn’t occur in \( A \) or \( B \)
Emulating Hilbert’s epsilon in ZFC (1 of 2)

The class expression “εxϕ” denotes “some x satisfying wff ϕ.” The **Transfinite Axiom** is a conservative extension of ZFC:

ϕ → [εxϕ/x]ϕ

where x is free in ϕ and [.../x]ϕ denotes proper substitution.

To emulate the transfinite axiom in ZFC, we define two class expressions A and B, where y is does not occur in ϕ:

\[
A = \{ x | (\varphi \land \forall y ([y/x]\varphi \rightarrow (\text{rank}'x) \subseteq (\text{rank}'y))) \}
\]

\[
B = \bigcup \{ x \in A | \forall y \in A \neg y r x \}
\]

**Theorem (hta in set.mm):**

r We A → (ϕ → [B/x]ϕ)

Class B emulates Hilbert’s epsilon εxϕ.
Emulating Hilbert’s epsilon in ZFC (2 of 2)

Epsilon-calculus proof

ϕ → [εxϕ/x]ϕ

(manipulate εxϕ)

(εxϕ-free result)

ZFC proof

r We A → (ϕ → [B(r)/x]ϕ)

(manipulate B(r))

r We A → (B(r)-free result)

∃rr We A → (B(r)-free result)

(B(r)-free result)

More details: